《片上存储器原型设计》

[《片上存储器原型设计》 1](#_Toc181356509)

[1. 第一节：On-chip-memery 3](#_Toc181356510)

[1.1ON-chip-memery 3](#_Toc181356511)

[1.1.1On-chip-memery分类 3](#_Toc181356512)

[1.1.2On-chip-memery标准参考接口 4](#_Toc181356513)

[1.2fifo 4](#_Toc181356514)

[1.2.1FIFO应用场景 5](#_Toc181356515)

[第二节：FIFO原型设计 7](#_Toc181356516)

[2.1架构设计 7](#_Toc181356517)

[2.2关于读写状态判断 7](#_Toc181356518)

[2.3存储器读写程序设计 8](#_Toc181356519)

[2.4写地址计算与状态判断 8](#_Toc181356520)

[2.5读地址计算与状态判断 9](#_Toc181356521)

# 第一节：On-chip-memery

存储器是数字电路设计中必不可少的电路，我们的**设计电路的核心是数据**，任何情况下数据始终是我们的电路设计的要素，只要有数据，就会存有储电路参与，对于数字电路设计者而言，**熟悉掌握存储器的应用是基本技能**。

从数据电路设计的应用角度来说，存储器分为片上存储器（on-chip-memery）和片外存储器。

On-chip-memery：存储器被集成在设计电路中，对于FPGA而言，存储器在其内部，对于ASIC而言，存储器被集成在设计电路中。

优点：读写数据延迟小，随时读写，读写操作简单

缺点：由于是作为设计电路的一部分，因此占用设计资源，存储量非常小

片外存储器 :存储器独立于设计电路，2个独立电路需要特定的引脚连接

优点 ：存储量大，可以存储大批量数据

缺点 ：读写延迟较大，读写操作复杂

## 1.1ON-chip-memery

On-chip-memery由于作为设计电路一部分，占用设计资源，因此存储数据量非常小，如果存储数据量较大，不适合用片上存储器。

在实际应用选中，片上存储器在设计中用于**缓存**，而不是用于**数据存储器**。

缓存 ：零时存储器，一般加快数据读写

数据存储器 ：真正用来存储数据

应用中存储器 = 缓存 + 数据存储器

### 1.1.1On-chip-memery分类







单端口ram

标准ram

单端口rom

标准rom

Ram

1:写操作

2:读操作

Rom

1:初始化操作

2:读操作

双端口rom

多端口rom

Ocm



移位存储器

fifo

双端口ram

多端口ram

Rom：一般用于lut(查找表)/复杂译码电路

Ram: 一般缓存数据，随机读写用双端口ram,顺序读写用fifo

### 1.1.2On-chip-memery标准参考接口

**写操作接口**

wr\_clk :写时钟

wr\_en :写使能

wrdata[W-1:0] :写数据

**读操作接口**

rd\_clk :读时钟

rd\_en :读使能，有时可以忽略

rddata[W-1:0] :读数据

rdaddr[A-1:0]/读存储状态标志 :写地址/读状态标志

### 1.2fifo

这里重点介绍FIFO，在数字电路设计中FIFO是On-chip-memery中使用率最高的，它是双端口ram的一种特殊结构，它和双端口ram的本质区别是顺序读写。

对于ram的2种读写方式，有着本质的区别，操作要求如下：

随机读写：

写数据：往指定的地址空间写入数据，当我们往同一地址空间写数据，之后的数据会覆盖掉之前的数据。

读数据：从指定的地址空间读出数据，当我们从同一地址空间读取数据，数据会反复被读出

**顺序读写：**

**写数据 ：从开始地址空间按顺序写数据，同一地址空间只能写一次数据，不支持数据覆盖**

**读数据 :从开始的地址空间按书序读出数据，同一地址空间只能读出一次数据（必须有有效数），不支持反复读数据**。

### 1.2.1FIFO应用场景

1. 跨域处理

电路系统2

电路系统1

Din[K0-1:0]

Dout[K1-1:0]

组合电路

组合电路

Clk2

Clk1

Clk1和clk2是2个不同的时钟，有可能是同步时钟，也有可能是异步时钟，我们把这2个电路系统叫做跨时钟域系统。

如果clk1和clk2是相关时钟，这2个系统为同步跨时钟域，否则异步跨时钟域。

无论是同步跨时钟域还是异步跨时钟域，为了保证2个电路系统能够稳定采样到数据，需要在这2个时钟域之间加缓存。

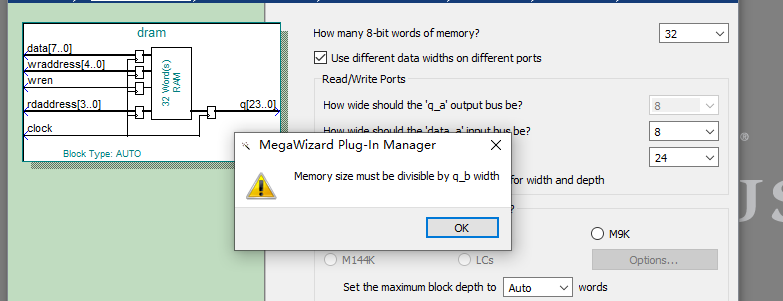
1. 带宽转换

现在片上存储器读写支持带宽转换，但是数据位宽之间的比例必须是2n倍的关系,满足如下关系：

K为2n是

Wrdata[k-1:0] rddata[N\*K-1:0]

rddata[k-1:0] wrdata[N\*K-1:0]



1. 缓存

在数字电路设计中任何情况都必须保证总带宽匹配（读带宽>=写带宽），当总带宽匹配的情况下保证了数据的稳定性，但是即使总带宽匹配了，有可能会存在瞬时带宽不匹配，数据稳定性不能完全保证。

总带宽 ：是数字电路设计的关键指标，任何电路保证总带宽匹配，在我们规定时间时间内数据传输的带宽。

**700bit**

200bit

500bit

120bit

T

T

T

t

t

T

T

T

T

**700bit**

200bit

500bit

120bit

电路总带宽 = XBITMAX/T (Xbit表示时间轴上T时间内最大数据量

T:突发周期)

瞬时带宽：在突发周期内某一时间 的带乱

T

在T0时刻的带宽

T0

最小深度计算

burst\_length ：突发数据个数  
X,Y：读时钟周期里，每Y个时钟周期会有X个数据读出FIFO  
r\_clk：读时钟  
w\_clk：写时钟

读数据有效频率= r\_clk\*（X/Y）

写时钟周期 =1/w\_clk

一个写周期内可以读出的数据个数 = (1/w\_clk \* r\_clk\*（X/Y）)

在突发周期内读出的数据个数 = bl\*(1/w\_clk \* r\_clk\*（X/Y）)

在突发周期内未读出的数据个数 = bl- bl\*(1/w\_clk \* r\_clk\*（X/Y）)

= bl- bl\*( r\_clk /w\_clk \*（X/Y）)

Fifo最小深度 = 2\*bl- 2\*bl\*( r\_clk /w\_clk \*（X/Y）)

# 第二节：FIFO原型设计

需求：

1. 不支持带宽转换
2. 数据位宽8位
3. 深度为16

## 2.1架构设计

rst\_n

rst\_n

rd\_en

rd\_clk

wr\_en

wr\_clk

rd\_en

rd\_clk

wr\_en

wr\_clk

mem\_rw

读写存储器

fifo

empty

empty

full

full

Wraddr\_gen\_full

计算写地址

判断写状态

wraddr\_e[4:0]

rdaddr\_e[4:0]

rdaddr[3:0]

wraddr[3:0]

rddata[7:0]

Wrdata[7:0]

rdaddr\_gen\_enpty

计算读地址

判断读状态

## 2.2关于读写状态判断

|  |
| --- |
| A0 |
| A1 |
| A2 |
| A3 |
| A4 |
| A… |
| An-1 |

读操作

写操作

## 2.3存储器读写程序设计

rd\_clk

Wrdata[7:0]

wr\_en

wr\_clk

empty

full

rdaddr[3:0]

wraddr[3:0]

mem\_rw

读写存储器

rddata[7:0]

功能：存储块读写

rd\_en

|  |  |
| --- | --- |
| 信号名 | 描述 |
| 写数据相关 | |
| wr\_clk | 写时钟 |
| wr\_en | 写使能 |
| wrdata[7:0] | 写数据 |
| wraddr[3:0] | 写地址 |
| full | 写状态标志 |
| 读数据相关 |  |
| rd\_clk | 读时钟 |
| rd\_en | 读使能 |
| rddata[7:0] | 读数据 |
| rdaddr[3:0] | 读地址 |
| empty | 读状态标志 |

## 2.4写地址计算与状态判断

功能：计算写地址，判断写状态

rst\_n

wr\_en

wr\_clk

full

wraddr[3:0]

Wraddr\_gen\_full

计算写地址

判断写状态

rdaddr\_e[4:0]

wraddr\_e[4:0]

|  |  |
| --- | --- |
| 信号名 | 描述 |
| 系统相关 | |
|  | 系统复位 |
| 写操作相关 | |
| wr\_en | 写使能 |
| wr\_clk | 写时钟 |
| 输出写地址与状态 | |
| wraddr[3:0] | 存储块写地址 |
| full | 当full==1表示满 |
| 存储块状态判断有关地址 | |
| wraddr\_e[4:0] | 扩位后写地址 |
| rdaddr\_e[4:0] | 扩位后读地址 |

## 2.5读地址计算与状态判断

功能：计算存储块读地址，判断状态

rd\_en

rd\_clk

empty

rdaddr[3:0]

rdaddr\_gen\_enpty

计算读地址

判断读状态

rst\_n

wraddr\_e[4:0]

rdaddr\_e[4:0]

|  |  |
| --- | --- |
| 信号名 | 描述 |
| 系统相关 | |
| rst\_n | 系统复位 |
| 写操作相关 | |
| rd\_en | 读使能 |
| rd\_clk | 读时钟 |
| 输出写地址与状态 | |
| rdaddr[3:0] | 存储块读地址 |
|  | 当empty ==1表示空 |
| 存储块状态判断有关地址 | |
| wraddr\_e[4:0] | 扩位后写地址 |
| rdaddr\_e[4:0] | 扩位后读地址 |